Appendix C. Code Listings

This appendix contains the source code to the many example programs developed throughout the book. To explain the programs, we often presented them piece by piece, sometimes omitting repetitive sections. Example C-1 through Example C-54 are the full programs: intact, unabridged, in all their glory. You can also download them from the web at http://www.oreilly.com/catalog/progwebsoap/.

C.1 Hello World in Perl

Example C-1. HelloWorld.pm (server)

```
package Hello;
sub sayHello {
  shift;
  my $self = "Hello " . shift;
}
1;
```

Example C-2. HelloWorld.cgi (server)

```
use SOAP::Transport::HTTP;
SOAP::Transport::HTTP::CGI
 -> dispatch_to('Hello::(?:sayHello)')
 -> handle
;
```

Example C-3. HelloWorldClient.pm (client)

```
use SOAP::Lite;
my $name = shift;
print "\n\nCalling the SOAP Server to say hello\n\n";
print "The SOAP Server says: ";
print SOAP::Lite
  -> uri('urn:Example1')
  -> proxy('http://localhost/cgi-bin/helloworld.cgi')
  -> sayHello($name)
  -> result . "\n\n";
```

C.2 Hello World Client in Visual Basic

Example C-4. Helloworld.vbs (client)

```
Dim x, h
Set x = CreateObject("MSXML2.DOMDocument")

x.loadXML "<s:Envelope  8
  xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'8
  xmlns:xsi='http://www.w3.org/1999/XMLSchema-instance'   8
  xmlns:xsd='http://www.w3.org/1999/XMLSchema'><s:Body><m:sayHello 8
  xmlns:m='urn:Example1'><name xsi:type='xsd:string'>James</name> 8
  </m:sayHello></s:Body></s:Envelope>"8

msgbox x.xml, , "Input SOAP Message"
Set h = CreateObject("Microsoft.XMLHTTP")
```

```
h.open "POST", "http://localhost/cgi-bin/helloworld.cgi"
h.send (x)
while h.readyState <> 4
wend
msgbox h.responseText,,"Output SOAP Message"
```

C.3 Hello World over Jabber

Example C-5. HelloWorldJabber.pm (server)

```
use SOAP::Transport::JABBER;
my $server = SOAP::Transport::JABBER::Server
  -> new('jabber://soaplite_server:soapliteserver@jabber.org:5222')
  -> dispatch_to('Hello')
;
print "SOAP Jabber Server Started\n";
do { $server->handle } while sleep 1;
```

Example C-6. HelloWorldJabberClient.pm (client)

```
use SOAP::Lite;
my $name = shift;
print "\n\nCalling the SOAP Server to say hello\n\n";
print "The SOAP Server says: ";
print SOAP::Lite
  -> uri('urn:Example1')
  -> proxy('jabber://soaplite_client:soapliteclient@jabber.org:5222/' .
           'soaplite_server@jabber.org/')
  -> sayHello($name)
  -> result . "\n\n";
```

C.4 Hello World in Java

Example C-7. Hello.java (server)

```
package samples;
public class Hello {
   public String sayHello(String name) {
       return "Hello " + name;
   }
}
```

Example C-8. Hello.java Deployment Descriptor (server)

```
<dd:service             xmlns:dd="http://xml.apache.org/xml-soap/deployment"
id="urn:Example1">
 <dd:provider type="java"
              scope="Application"
              methods="sayHello">
    <dd:java class="samples.Hello"
             static="false" />
 </dd:provider>
 <dd:faultListener>
  org.apache.soap.server.DOMFaultListener
 </dd:faultListener>
 <dd:mappings />
</dd:service>
```

Example C-9. Hello_Client.java (client)

```
import java.io.*;
import java.net.*;
import java.util.*;
import org.apache.soap.*;
import org.apache.soap.rpc.*;

public class Hello_client {

  public static void main (String[] args)
       throws Exception {

    System.out.println("\n\nCalling the SOAP Server to say hello\n\n");
    URL url = new URL (args[0]);
    String name = args[1];

    Call call = new Call ();
    call.setTargetObjectURI("urn:Example1");
    call.setMethodName("sayHello");
    call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC;);
    Vector params = new Vector ();
    params.addElement (new Parameter("name", String.class, name, null));
    call.setParams (params);

    System.out.print("The SOAP Server says: ");

    Response resp = call.invoke(url, "");

    if (resp.generatedFault ()) {
      Fault fault = resp.getFault ();
      System.out.println ("\nOuch, the call failed: ");
      System.out.println ("  Fault Code   = " + fault.getFaultCode ());
      System.out.println ("  Fault String = " + fault.getFaultString ());
    } else {
      Parameter result = resp.getReturnValue ();
      System.out.print(result.getValue ());
      System.out.println();
    }
  }
}
```

C.5 Hello, World in C# on .NET

Example C-10. Helloworld.asmx (server)

```
<%@ WebService Language="C#" Class="Example1" %>

using System.Web.Services;

[WebService(Namespace="urn:Example1")]
public class Example1 {

    [ WebMethod ]
    public string sayHello(string name) {
        return "Hello " + name;
    }

}
```

Example C-11. Helloworld.cs (client)

```
// HelloWorld.cs

using System.Diagnostics;
using System.Xml.Serialization;
using System;
using System.Web.Services.Protocols;
using System.Web.Services;

[System.Web.Services.WebServiceBindingAttribute(
     Name="Example1Soap",
     Namespace="urn:Example1")]
public class Example1 :
             System.Web.Services.Protocols.SoapHttpClientProtocol {

    public Example1() {
        this.Url = "http://localhost/helloworld.asmx ";
    }

    [System.Web.Services.Protocols.SoapDocumentMethodAttribute(
         "urn:Example1/sayHello",
         RequestNamespace="urn:Example1",
         ResponseNamespace="urn:Example1",
         Use=System.Web.Services.Description.SoapBindingUse.Literal,

ParameterStyle=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)]
    public string sayHello(string name) {
        object[] results = this.Invoke("sayHello",
                                       new object[] {name});
        return ((string)(results[0]));
    }

    public static void Main(string[] args) {
       Console.WriteLine("Calling the SOAP Server to say hello");
       Example1 example1 = new Example1();
       Console.WriteLine("The SOAP Server says: " +
                         example1.sayHello(args[0]));
    }
}
```

Example C-12. Modified Perl HelloWorld_Client.pm for use with .NET (client)

```
use SOAP::Lite;

my $name = shift;

print "\n\nCalling the SOAP Server to say hello\n\n";

print "The SOAP Server says: ";

print SOAP::Lite
  -> uri('urn:Example1')
  -> on_action(sub{sprintf '%s/%s', @_ })
  -> proxy('http://localhost:8080/helloworld/example1.asmx')
  ->     sayHello(SOAP::Data->name(name     =>     $name->type->('string')-
>uri('urn:Example1'))
  -> result . "\n\n";
```

C.6 Publisher Service

Example C-13. Publisher.pm (server)

```
package Publisher;

use strict;

package Publisher::DB;

use DBI;
use vars qw($CONNECT);

$CONNECT = "DBI:CSV:f_dir=/home/soaplite/book;csv_sep_char=\0";

my $dbh;

sub dbh {
  shift;
  unless ($dbh) {
    $dbh = DBI->connect(shift || $CONNECT);
    $dbh->{'RaiseError'} = 1;
  }
  return $dbh;
}

END { $dbh->disconnect if $dbh; }

sub create {
  my $dbh = shift->dbh;

  $dbh->do($_) foreach split /;/, '

  CREATE TABLE members (
    memberID   integer,
    email      char(100),
    password   char(25),
    firstName  char(50),
    lastName   char(50),
    title      char(50),
    company    char(50),
    url        char(255),
    subscribed integer
  );

  CREATE TABLE items (
    itemID     integer,
    memberID   integer,
    type        integer,
    title      char(255),
    description char(512),
    postStamp  integer
  )

';

}
```

```
sub insert_member {
  my $dbh = shift->dbh;
  my $newMemberID = 1 + $dbh->selectrow_array(
         "SELECT memberID FROM members ORDER BY memberID DESC");

  my %parameters = (@_, memberID => $newMemberID, subscribed => 0);
  my $names = join ', ', keys %parameters;
  my $placeholders = join ', ', ('?') x keys %parameters;

  $dbh->do("INSERT INTO members ($names) VALUES ($placeholders)", {},
           values %parameters);
  return $newMemberID;
}

sub select_member {
  my $dbh = shift->dbh;
  my %parameters = @_;

  my $where = join ' AND ', map {"$_ = ?"} keys %parameters;
  $where = "WHERE $where" if $where;

  # returns row in array context and first element (memberID) in scalar
  return $dbh->selectrow_array("SELECT * FROM members $where", {},
                               values %parameters);
}

sub update_member {
  my $dbh = shift->dbh;
  my($memberID, %parameters) = @_;

  my $set = join ', ', map {"$_ = ?"} keys %parameters;

  $dbh->do("UPDATE members SET $set WHERE memberID = ?", {},
           values %parameters, $memberID);
  return $memberID;
}

sub insert_item {
  my $dbh = shift->dbh;
  my $newItemID = 1 + $dbh->selectrow_array(
            "SELECT itemID FROM items ORDER BY itemID DESC");

  my %parameters = (@_, itemID => $newItemID, postStamp => time());
  my $names = join ', ', keys %parameters;
  my $placeholders = join ', ', ('?') x keys %parameters;

  $dbh->do("INSERT INTO items ($names) VALUES ($placeholders)", {},
           values %parameters);

  return $newItemID;
}

sub select_item {
  my $dbh = shift->dbh;
  my %parameters = @_;

  my $where = join ' AND ', map {"$_ = ?"} keys %parameters;

  return $dbh->selectrow_array("SELECT * FROM items WHERE $where", {},
                               values %parameters);
}
```

```
sub select_all_items {
  my $dbh = shift->dbh;
  my %parameters = @_;

  my $where = join ' AND ', map {"$_ = ?"} keys %parameters;
  $where = "WHERE $where" if $where;

  return $dbh->selectall_arrayref("SELECT type, title, description,
      postStamp, memberID FROM items $where", {}, values %parameters);
}

sub delete_item {
  my $dbh = shift->dbh;
  my $itemID = shift;

  $dbh->do('DELETE FROM items WHERE itemID = ?', {}, $itemID);
  return $itemID;
}

# ====================================================================

package Publisher;

use POSIX qw(strftime);

@Publisher::ISA = qw(SOAP::Server::Parameters);

# --------------------------------------------------------------------
# private functions
# --------------------------------------------------------------------

use Digest::MD5 qw(md5);

my $calculateAuthInfo = sub {
  return md5(join '', 'unique (yet persistent) string', @_);
};

my $checkAuthInfo = sub {
  my $authInfo = shift;
  my $signature = $calculateAuthInfo->(@{$authInfo}{qw(memberID email
                                        time)});
  die "Authentication information is not valid\n" if $signature ne
        $authInfo->{signature};
  die "Authentication information is expired\n"
  if time() > $authInfo->{time};
  return $authInfo->{memberID};
};

my $makeAuthInfo = sub {
  my($memberID, $email) = @_;
  my $time = time()+20*60;
  my $signature = $calculateAuthInfo->($memberID, $email, $time);
  return +{memberID => $memberID, time => $time, email => $email,
        signature => $signature};
};
```

```
# ------------------------------------------------------------------------
# public functions
# ------------------------------------------------------------------------

sub register {
  my $self = shift;
  my $envelope = pop;
  my %parameters = %{$envelope->method() || {}};

  die "Wrong parameters: register(email, password, firstName, lastName [,
       title][, company][, url])\n"
    unless 4 == map {defined} @parameters{qw(email password firstName
                lastName)};

  my $email = $parameters{email};
  die "Member with email ($email) already registered\n"
    if Publisher::DB->select_member(email => $email);
  return Publisher::DB->insert_member(%parameters);
}

sub modify {
  my $self = shift;
  my $envelope = pop;
  my %parameters = %{$envelope->method() || {}};

  my $memberID = $checkAuthInfo->($envelope->valueof('//authInfo'));
  Publisher::DB->update_member($memberID, %parameters);
  return;
}

sub login {
  my $self = shift;
  my %parameters = %{pop->method() || {}};

  my $email = $parameters{email};
  my $memberID = Publisher::DB->select_member(email => $email,
                                              password                    =>
$parameters{password});
  die "Credentials are wrong\n" unless $memberID;
  return bless $makeAuthInfo->($memberID, $email) => 'authInfo';
}

sub subscribe {
  my $self = shift;
  my $memberID = $checkAuthInfo->(pop->valueof('//authInfo'));

  Publisher::DB->update_member($memberID, subscribed => 1);
  return;
}

sub unsubscribe {
  my $self = shift;
  my $memberID = $checkAuthInfo->(pop->valueof('//authInfo'));

  Publisher::DB->update_member($memberID, subscribed => 0);
  return;
}

my %type2code = (news => 1, article => 2, resource => 3);
my %code2type = reverse %type2code;
```

```
sub postItem {
  my $self = shift;
  my $envelope = pop;
  my $memberID = $checkAuthInfo->($envelope->valueof('//authInfo'));
  my %parameters = %{$envelope->method() || {}};

  die "Wrong parameter(s): postItem(type, title, description)\n"
    unless 3 == map {defined} @parameters{qw(type title description)};

  $parameters{type} = $type2code{lc $parameters{type}}
                                  or    die    "Wrong    type    of    item
($parameters{type})\n";
  return Publisher::DB->insert_item(memberID => $memberID, %parameters);
}

sub removeItem {
  my $self = shift;
  my $memberID = $checkAuthInfo->(pop->valueof('//authInfo'));
  die "Wrong parameter(s): removeItem(itemID)\n" unless @_ == 1;

  my $itemID = shift;
  die "Specified item ($itemID) can't be found or removed\n"
    unless  Publisher::DB->select_item(memberID  =>  $memberID,  itemID  =>
$itemID);
  Publisher::DB->delete_item($itemID);
  return;
}

my $browse = sub {
  my $envelope = pop;
  my %parameters = %{$envelope->method() || {}};

  my($type, $format, $maxRows, $query) = @parameters{qw(type format maxRows
query)};
  $type = {all => 'all', %type2code}->{lc($type) || 'all'} or
    die "Wrong type of item ($type)\n";

  $maxRows ||= 25;
  $format ||= 'XML';
  my  $items  =  Publisher::DB->select_all_items($type  ne  'all'  ?  (type  =>
$type) : ());
  my %members;
  my @items = map {
    my($type, $title, $description, $date, $memberID) = @$_;
    my($email, $firstName, $lastName) = @{
      $members{$memberID}  ||=  [Publisher::DB->select_member(memberID  =>
$memberID)]
    }[1,3,4];
    +{
      $format =~ /^XML/ ? (
        type        => $code2type{$type},
        title       => $title,
        description => $description,
        date        => strftime("%Y-%m-%d", gmtime($date)),
        creator     => "$firstName $lastName ($email)"
      ) : (
        category    => $code2type{$type},
        title       => "$title by $firstName $lastName ($email) on "
                       . strftime("%Y-%m-%d", gmtime($date)),
        description => $description,
      )
```

```
    }
  }  @{$items}[0..(!$query  &&  $maxRows  <=  $#$items  ?  $maxRows-1  :
$#$items)];
  if ($query) {
    my $regexp = join '', map {
      /\s+and\s+/io ? '&&' : /\s+or\s+/io ? '||' : /[()]/ ? $_ : $_ ? '/' .
quotemeta($_) . '/o' : ''
    } split /(\(|\)|\s+and\s+|\s+or\s+)/io, $query;
    eval "*checkfor = sub { for (\@_) { return 1 if $regexp; } return }" or
die;
    @items = grep {checkfor(values %$_)} @items;
    splice(@items, $maxRows <= $#items ? $maxRows : $#items+1);
  }
  return $format =~ /^(XML|RSS)str$/
    ? SOAP::Serializer
        -> autotype(0)
        -> readable(1)
        ->  serialize(SOAP::Data->name(($1  eq  'XML'  ?  'itemList'  :
'channel')
                      => \SOAP::Data->name(item => @items)))
    : [@items];
};

sub browse {
  my $self = shift;
  return SOAP::Data->name(browse => $browse->(@_));
}

sub search {
  my $self = shift;
  return SOAP::Data->name(search => $browse->(@_));
}

# ========================================================================

1;
```

Example C-14. Publisher.daemon (server)

```
#!/bin/perl

use SOAP::Transport::HTTP;

use Publisher;

$Publisher::DB::CONNECT =
   "DBI:CSV:f_dir=d:/book;csv_sep_char=\0";
$authinfo = 'http://www.soaplite.com/authInfo';

my $server = SOAP::Transport::HTTP::CGI
  -> dispatch_to('Publisher');

$server->serializer->maptype({authInfo => $authinfo});
$server->handle;
```

Example C-15. Client.java (client)

```
import java.io.*;
import java.net.*;
import java.util.*;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.*;

import org.apache.soap.util.xml.*;
import org.apache.soap.*;
import org.apache.soap.encoding.*;
import org.apache.soap.encoding.soapenc.*;
import org.apache.soap.rpc.*;

public class Client {

  private URL url;
  private String uri;
  private authInfo authInfo;

  public Client (String url, String uri) throws Exception {
    try {
      this.uri = uri;
      this.url = new URL(url);
    } catch (Exception e) {
      throw new Exception(e.getMessage());
    }
  }

  public Header makeAuthHeader (authInfo auth) throws Exception {
    if (auth == null) {
        throw new Exception(
            "Oops, you are not logged in. Please login first"); }
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    dbf.setNamespaceAware(true);
    dbf.setValidating(false);
    DocumentBuilder db = dbf.newDocumentBuilder();
    Document doc = db.newDocument();
    Element authEl =
         doc.createElementNS("http://www.soaplite.com/authInfo",
                             "auth:authInfo");
    Element emailEl = doc.createElement("email");
    emailEl.appendChild(doc.createTextNode(auth.getEmail()));
    Element signatureEl = doc.createElement("signature");
    signatureEl.setAttribute("xmlns:enc", Constants.NS_URI_SOAP_ENC);
    signatureEl.setAttribute("xsi:type", "enc:base64");
    signatureEl.appendChild(doc.createTextNode(
        Base64.encode(auth.getSignature())));
    Element memberIdEl = doc.createElement("memberID");
    memberIdEl.appendChild(doc.createTextNode(String.valueOf(
        auth.getMemberID())));
    Element timeEl = doc.createElement("time");
    timeEl.appendChild(doc.createTextNode(String.valueOf(
        auth.getTime())));
    authEl.appendChild(emailEl);
    authEl.appendChild(signatureEl);
    authEl.appendChild(memberIdEl);
    authEl.appendChild(timeEl);
    Vector headerEntries = new Vector();
    headerEntries.add(authEl);
    Header header = new Header();
    header.setHeaderEntries(headerEntries);
    return header;
  }
```

```
private Call initCall () {
  Call call = new Call();
  call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);
  call.setTargetObjectURI(uri);
  return call;
}

private Object invokeCall (Call call) throws Exception {
  try {
    Response response = call.invoke(url, "");

    if (!response.generatedFault()) {
      return response.getReturnValue() == null
        ? null : response.getReturnValue().getValue();
    } else {
      Fault f = response.getFault();
      throw new Exception("Fault = " + f.getFaultCode() + ", " +
                          f.getFaultString());
    }
  } catch (SOAPException e) {
    throw new Exception("SOAPException = " + e.getFaultCode() + ", " +
                        e.getMessage());
  }
}

public void login (String email, String password) throws Exception {
  Call call = initCall();

  SOAPMappingRegistry smr = new SOAPMappingRegistry();
  BeanSerializer beanSer = new BeanSerializer();
  smr.mapTypes(Constants.NS_URI_SOAP_ENC,
               new QName("http://www.soaplite.com/Publisher",
                         "authInfo"),
               authInfo.class, beanSer, beanSer);

  Vector params = new Vector ();
  params.add(new Parameter("email", String.class, email, null));
  params.add(new Parameter("password", String.class, password, null));
  call.setParams(params);
  call.setMethodName("login");
  call.setSOAPMappingRegistry(smr);

  authInfo = (authInfo) invokeCall(call);

  System.out.println(authInfo.getEmail() + " logged in.");
}

public void register (String email, String password,
                      String firstName, String lastName,
                      String title, String company, String url)
                      throws Exception {
  Call call = initCall();

  Vector params = new Vector ();
  params.add(new Parameter("email", String.class, email, null));
  params.add(new Parameter("password", String.class, password, null));
  params.add(new Parameter("firstName", String.class, firstName, null));
  params.add(new Parameter("lastName", String.class, lastName, null));
  if (url != null)
      params.add(new Parameter("url", String.class, url, null));
```

```
  if (title != null)
      params.add(new Parameter("title", String.class, title, null));
  if (company != null)
      params.add(new Parameter("company", String.class, company, null));
  call.setParams(params);
  call.setMethodName("register");
  invokeCall(call);
  System.out.println("Registered.");
}

public void postItem (String type, String title,
                      String description)
                      throws Exception {
  Call call = initCall();
  Vector params = new Vector ();
  params.add(new Parameter("type", String.class, type, null));
  params.add(new Parameter("title", String.class, title, null));
  params.add(new Parameter("description", String.class, description,
                           null));
  call.setParams(params);
  call.setMethodName("postItem");
  call.setHeader(makeAuthHeader(authInfo));
  Integer itemID = (Integer)invokeCall(call);
  System.out.println("Posted item " + itemID + ".");
}

public void removeItem (Integer itemID) throws Exception {
  Call call = initCall();
  Vector params = new Vector ();
  params.add(new Parameter("itemID", Integer.class, itemID, null));
  call.setParams(params);
  call.setMethodName("removeItem");
  call.setHeader(makeAuthHeader(authInfo));
  invokeCall(call);
  System.out.println("Removed item " + itemID + ".");
}

public void browse (String type, String format,
                    Integer maxRows)
                    throws Exception {
  Call call = initCall();
  Vector params = new Vector ();
  params.add(new Parameter("format", String.class, format != null ?
             format : "XMLstr", null));
  if (type != null)    params.add(new Parameter("type", String.class,
             type, null));
  if (maxRows != null) params.add(new Parameter("maxRows",
             Integer.class, maxRows, null));
  call.setParams(params);
  call.setMethodName("browse");
  System.out.println((String)invokeCall(call));
}

public static void main(String[] args) {

  String myname = Client.class.getName();
  if (args.length < 1) {
    System.err.println("Usage:\n  java " + myname +
      " SOAP-router-URL");
    System.exit (1);
  }
```

```
    try {
      Client client = new Client(args[0],
                     "http://www.soaplite.com/Publisher");

      InputStream in = System.in;
      InputStreamReader isr = new InputStreamReader(in);
      BufferedReader br = new BufferedReader(isr);
      String action = null;
      while (!("quit".equals(action))) {
         System.out.print("> ");
         action = br.readLine();


         if ("register".equals(action)) {

             String email = null;
             String password = null;
             String firstName = null;
             String lastName = null;
             String title = null;
             String company = null;
             String url = null;

             System.out.print("\n\nIn order to register, you must answer
the following questions.");
             System.out.print("\n\nWhat is your email address: ");
             email = br.readLine();
             System.out.print("\nWhat is your first name: ");
             firstName = br.readLine();
             System.out.print("\nWhat is your last name: ");
             lastName = br.readLine();
             System.out.print("\nWhat is your job title: ");
             title = br.readLine();
             System.out.print("\nWhat company do you work for: ");
             company = br.readLine();
             System.out.print("\nWhat is your company or personal URL: ");
             url = br.readLine();
             System.out.print("\nFinally, what password do you want to use:
");
             password = br.readLine();

             System.out.println("\nAttempting to register....");
             client.register(email, password, firstName,
                             lastName, title, company, url);
             System.out.println();
         }

         if ("login".equals(action)) {
             String id = null;
             String pwd = null;

             System.out.print("\n\nWhat is your user id: ");
             id = br.readLine();
             System.out.print("\nWhat is your password: ");
             pwd = br.readLine();

             System.out.println("\nAttempting to login....");
             client.login(id,pwd);
             System.out.println();
         }
```

```
        if ("post".equals(action)) {

            String type = null;
            String title = null;
            String desc = null;

            System.out.print("\n\nWhat  type  of  item  [1  =  News,  2  =
Article,
            3 = Resource]: ");
            type = br.readLine();
            if (type.equals("1")) type = "news";
            if (type.equals("2")) type = "article";
            if (type.equals("3")) type = "resource";
            System.out.println("\nWhat is the title: ");
            title = br.readLine();
            System.out.println("\nWhat is the description: ");
            desc = br.readLine();

            System.out.println("\nAttempting to post item....");
            client.postItem(type, title, desc);
            System.out.println();
        }

        if ("remove".equals(action)) {
            System.out.print("\n\nPlease enter the numeric ID of the item
to remove: ");
            String id = br.readLine();
            try {
               System.out.println("\nAttempting to remove item....");
               client.removeItem(Integer.valueOf(id));
            } catch (Exception ex) {
               System.out.println("\nCould not remove item!");
            }
            System.out.println();
        }

        if ("browse".equals(action)) {
            System.out.print("\n\nWhat is the maximum number of rows
            to return
(blank to return all): ");
            String mRows = br.readLine();
            System.out.print("\nType of resource to browse ([0] = All, [1]
= News,
            [2] = Article, [3] = Resource): ");
            String type = br.readLine();
            if (type.equals("0")) type = "all";
            if (type.equals("1")) type = "news";
            if (type.equals("2")) type = "article";
            if (type.equals("3")) type = "resource";
            System.out.print("\nHow would you like to see the results ([1]
= XML,
            [2] = RSS): ");
            String format = br.readLine();
            if (format.equals("1")) format = "XMLstr";
            if (format.equals("2")) format = "RSSstr";

            System.out.println("\nAttempting to browse....");
            try {
                Integer ival = null;
                if (!("".equals(mRows))) {
                    ival = Integer.valueOf(mRows);
```

```
                }
                client.browse(type, format, ival);
            } catch (Exception ex) {
               System.out.println(ex);
               System.out.println("\nCould not browse!");
            }
         }

         if ("help".equals(action)) {
             System.out.println("\nActions:  register  |  login  |  post  |
remove | browse");
         }
       }
     } catch (Exception e) {
       System.err.println("Caught Exception: " + e.getMessage());
     }
   }
}
```

This is WSDL for the Hello World service.

Example C-16. WSDL for the Hello World service

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions   name="HelloWorld"
          targetNamespace="urn:HelloWorld"
          xmlns:tns="urn:HelloWorld"
          xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
          xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:message name="sayHello_IN">
    <part name="name" type="xsd:string" />
  </wsdl:message>
  <wsdl:message name="sayHello_OUT">
    <part name="greeting" type="xsd:string" />
  </wsdl:message>
  <wsdl:portType name="HelloWorldInterface">
    <wsdl:operation name="sayHello" >
      <wsdl:input message="tns:sayHello_IN" />
      <wsdl:output message="tns:sayHello_OUT" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="HelloWorldBinding"
    type="tns:HelloWorldInterface">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="sayHello">
     <soap:operation soapAction="urn:Hello" />
     <wsdl:input>
      <soap:body use="encoded"
        namespace="urn:Hello"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
     </wsdl:input>
     <wsdl:output>
      <soap:body use="encoded"
        namespace="urn:Hello"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
     </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="HelloWorldService">
```

```
    <wsdl:port name="Perl_HelloWorld" binding="tns:Binding_Name">
      <soap:address
        location="http://localhost/cgi-bin/hello.cgi" />
    </wsdl:port>
    <wsdl:port name="Java_HelloWorld" binding="tns:Binding_Name">
      <soap:address
        location="http://localhost:8080/soap/servlet/rpcrouter" />
    </wsdl:port>
    <wsdl:port name="NET_HelloWorld" binding="tns:Binding_Name">
      <soap:address
        location="http://localhost/helloworld.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Example C-17. Authinfo.java (client)

```
public class authInfo {
  private int memberID;
  private long time;
  private String email;
  private byte [] signature;

  public authInfo() { }

  public authInfo(int memberID, long time, String email, byte[] signature)
{
    this.memberID = memberID;
    this.time = time;
    this.email = email;
    this.signature = signature;
  }

  public void setMemberID(int memberID) {
    this.memberID = memberID;
  }

  public int getMemberID() {
    return memberID;
  }

  public void setTime(long time) {
    this.time = time;
  }
  public long getTime() {
    return time;
  }

  public void setEmail(String email) {
    this.email = email;
  }

  public String getEmail() {
    return email;
  }

  public void setSignature(byte [] signature) {
    this.signature = signature;
  }

  public byte [] getSignature() {
```

```
    return signature;
  }

  public String toString() {
    return "[" + memberID + "] " + email;
  }
}
```

C.7 SAML Generation

Example C-18. Assertion.java

```
package saml;

import java.util.Date;
import org.w3c.dom.Element;
import org.w3c.dom.Document;

public abstract class Assertion implements AssertionAbstractType {

    private IDType assertionID;
    private String issuer;
    private Date issueInstant;

    public String getVersion() {
        return "1.0";
    }

    public IDType getAssertionID() {
        return this.assertionID;
    }

    public void setAssertionID(IDType assertionID) {
        this.assertionID = assertionID;
    }

    public String getIssuer() {
        return this.issuer;
    }

    public void setIssuer(String issuer) {
        this.issuer = issuer;
    }

    public Date getIssueInstant() {
        return this.issueInstant;
    }

    public void setIssueInstant(Date issueInstant) {
        this.issueInstant = issueInstant;
    }

    protected void serializeAttributes(Element e) {
        e.setAttribute("Version", getVersion());
        if (assertionID != null)
            e.setAttribute("AssertionID", assertionID.getText());
        if (issuer != null)
            e.setAttribute("Issuer", issuer);
        if (issueInstant != null)
            e.setAttribute("IssueInstant", issueInstant.toString());
```

```
    }

    protected void deserializeAttributes(Element source) {
        String s1 = source.getAttribute("AssertionID");
        String s2 = source.getAttribute("Issuer");
        String s3 = source.getAttribute("IssueInstant");
        if (s1 != null) setAssertionID(new IDType(s1));
        if (s2 != null) setIssuer(s2);
        if (s3 != null) setIssueInstant(new Date(s3));
    }

    public abstract void serialize(Element parent);
}
```

Example C-19. AssertionAbstractType.java

```
package saml;

import java.util.Date;

public interface AssertionAbstractType {

    public String getVersion();
    public IDType getAssertionID();
    public void setAssertionID(IDType assertionID);
    public String getIssuer();
    public void setIssuer(String issuer);
    public Date getIssueInstant();
    public void setIssueInstant(Date issueInstant);
}
```

Example C-20. AssertionFactory.java

```
package saml;

import java.util.Date;

public class AssertionFactory {

    public static AuthenticationAssertion newInstance(String id,
                                                      String issuerName,
                                                      Date issueInstant,
                                                      String name,
                                                      String domain,
                                                      String method,
                                                      Date authInstant,
                                                      String ip,
                                                      String dns) {

        AuthenticationAssertion aa = new AuthenticationAssertion();
        IDType aid = new IDType(id);
        aa.setAssertionID(aid);
        aa.setIssuer(issuerName);
        aa.setIssueInstant(issueInstant);
        Subject subject = new Subject();
        {
            NameIdentifier ni = new NameIdentifier();
            ni.setName(name);
            ni.setSecurityDomain(domain);
            subject.setNameIdentifier(ni);
```

```
            aa.setSubject(subject);
        }
        aa.setAuthenticationMethod(new AuthenticationMethod(method));
        aa.setAuthenticationInstant(
            new AuthenticationInstant(authInstant));
        AuthenticationLocale locale = new AuthenticationLocale();
        locale.setIP(ip);
        locale.setDNSDomain(dns);
        aa.setAuthenticationLocale(locale);
        return aa;
    }

}
```

Example C-21. AssertionID.java

```
package saml;

import org.w3c.dom.Element;
import org.w3c.dom.Document;

public class AssertionID extends IDType {

    public AssertionID() {}

    public AssertionID(String value) { super(value); }

    public void serialize(Element parent) {
        Document doc = parent.getOwnerDocument();
        Element e = doc.createElementNS(SAMLUtil.NS, "AssertionID");
        e.appendChild(doc.createTextNode(getText()));
        parent.appendChild(e);
    }

    public void deserialize(Element source) {
        String id = SAMLUtil.getInnerText(source);
        setText(id);
    }
}
```

Example C-22. AssertionSigner.java

```
package saml;

import java.io.FileInputStream;
import java.security.InvalidKeyException;
import java.security.Key;
import java.security.KeyStore;
import java.security.KeyStoreException;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.SignatureException;
import java.security.UnrecoverableKeyException;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import com.ibm.xml.dsig.*;
import org.w3c.dom.*;
```

```
public class AssertionSigner {

    public static Element sign(AuthenticationAssertion assertion,
                               String keystorepath,
                               String alias,
                               String storepass,
                               String keypass)
                               throws Exception {

        Document doc = SAMLUtil.newDocument();
        Element root = doc.createElement("root");
        assertion.serialize(root);

        //** Prepare the signature **//
        SignatureGenerator siggen = new SignatureGenerator(doc,
                                    DigestMethod.SHA1,
                                    Canonicalizer.W3C,
                                    SignatureMethod.DSA, null);
        siggen.addReference(
            siggen.createReference(
                siggen.wrapWithObject(root.getFirstChild(),
                    assertion.getAssertionID().getText())
                )
            );


        //** Prepare the key **//
        KeyStore keystore = KeyStore.getInstance("JKS");
        keystore.load(new FileInputStream(keystorepath),
                      storepass.toCharArray());
        X509Certificate cert =
            (X509Certificate)keystore.getCertificate(alias);
        Key key = keystore.getKey(alias, keypass.toCharArray());
        if (key == null) {
            throw new IllegalArgumentException("Invalid Key Info");
        }
        KeyInfo keyInfo = new KeyInfo();
        KeyInfo.X509Data x5data = new KeyInfo.X509Data();
        x5data.setCertificate(cert);
        x5data.setParameters(cert, true, true, true);
        keyInfo.setX509Data(new KeyInfo.X509Data[] { x5data });
        keyInfo.setKeyValue(cert.getPublicKey());
        siggen.setKeyInfoGenerator(keyInfo);

        //** Sign it **//
        Element sig = siggen.getSignatureElement();
        SignatureContext context = new SignatureContext();
        context.sign(sig, key);
        return sig;
    }

}
```

Example C-23. AssertionSpecifier.java

```
package saml;

import org.w3c.dom.Element;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
```

```
public class AssertionSpecifier implements AssertionSpecifierType {

    private AssertionID assertionID;
    private Assertion assertion;

    public AssertionID getAssertionID() {
        return this.assertionID;
    }

    public void setAssertionID(AssertionID assertionID) {
        this.assertionID = assertionID;
    }

    public Assertion getAssertion() {
        return this.assertion;
    }

    public void setAssertion(Assertion assertion) {
        this.assertion = assertion;
    }

    public void serialize(Element parent) {
        Document doc = parent.getOwnerDocument();
        Element e = doc.createElementNS(SAMLUtil.NS, "AssertionSpecifier");
        if (assertionID != null) assertionID.serialize(e);
        if (assertion != null) assertion.serialize(e);
        parent.appendChild(e);
    }

    public void deserialize(Element source) {
        NodeList nl = source.getChildNodes();
        for (int n = 0; n < nl.getLength(); n++) {
            Node node = nl.item(n);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element e = (Element)node;
                if ("AssertionID".equals(e.getLocalName())) {
                    AssertionID aid = new AssertionID();
                    aid.deserialize(e);
                    setAssertionID(aid);
                }
                if ("AuthenticationAssertion".equals(e.getLocalName())) {
                    AuthenticationAssertion aa = new
AuthenticationAssertion();
                    aa.deserialize(e);
                    setAssertion(aa);
                }
            }
```

.....Example C-24. AssertionSpecifierType.java

```
package saml;

public interface AssertionSpecifierType {

    public AssertionID getAssertionID();
    public void setAssertionID(AssertionID assertionID);
    public Assertion getAssertion();
    public void setAssertion(Assertion assertion);

}
```

Example C-25. AuthenticationAssertion.java

```
package saml;

import java.util.Date;
import org.w3c.dom.Element;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;

public class AuthenticationAssertion
    extends SubjectAssertion implements AuthenticationAssertionType {

    private AuthenticationMethod method;
    private AuthenticationInstant instant;
    private AuthenticationLocale locale;

    public AuthenticationMethod getAuthenticationMethod() {
        return this.method;
    }

    public void setAuthenticationMethod(AuthenticationMethod method) {
        this.method = method;
    }

    public AuthenticationInstant getAuthenticationInstant() {
        return this.instant;
    }

    public void setAuthenticationInstant(AuthenticationInstant instant) {
        this.instant = instant;
    }

    public AuthenticationLocale getAuthenticationLocale() {
        return this.locale;
    }

    public void setAuthenticationLocale(AuthenticationLocale locale) {
        this.locale = locale;
    }

    public void serialize(Element parent) {
        Document doc = parent.getOwnerDocument();
        Element         e         =         doc.createElementNS(SAMLUtil.NS,
"AuthenticationAssertion");
        e.setAttribute("xmlns", SAMLUtil.NS);
        serializeAttributes(e);
        serializeSubject(e);
        if (method != null) method.serialize(e);
        if (instant != null) instant.serialize(e);
        if (locale != null) locale.serialize(e);
        parent.appendChild(e);
    }

    public void deserialize(Element source) {
        deserializeAttributes(source);
        NodeList nl = source.getChildNodes();
        for (int n = 0; n < nl.getLength(); n++) {
            Node node = nl.item(n);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element e = (Element)node;
```

```
                if ("Subject".equals(e.getLocalName())) {
                    Subject subject = new Subject();
                    subject.deserialize(e);
                    setSubject(subject);
                }
                if ("AuthenticationMethod".equals(e.getLocalName())) {
                    AuthenticationMethod method = new
AuthenticationMethod();
                    method.deserialize(e);
                    setAuthenticationMethod(method);
                }
                if ("AuthenticationInstant".equals(e.getLocalName())) {
                    AuthenticationInstant instant = new
AuthenticationInstant();
                    instant.deserialize(e);
                    setAuthenticationInstant(instant);
                }
                if ("AuthenticationLocale".equals(e.getLocalName())) {
                    AuthenticationLocale locale = new
AuthenticationLocale();
                    locale.deserialize(e);
                    setAuthenticationLocale(locale);
                }
            }
        }
    }
}
```

Example C-26. AuthenticationAssertionType.java

```
package saml;

import java.util.Date;

public interface AuthenticationAssertionType extends
SubjectAssertionAbstractType {

    public AuthenticationMethod getAuthenticationMethod();
    public void setAuthenticationMethod(AuthenticationMethod method);
    public AuthenticationInstant getAuthenticationInstant();
    public void setAuthenticationInstant(AuthenticationInstant instant);
    public AuthenticationLocale getAuthenticationLocale();
    public void setAuthenticationLocale(AuthenticationLocale locale);

}
```

Example C-27. AuthenticationInstant.java

```
package saml;

import java.util.Date;
import org.w3c.dom.Element;
import org.w3c.dom.Document;



public class AuthenticationInstant {

    private Date instant;
```

```
    public AuthenticationInstant() {}

    public AuthenticationInstant(Date instant) {
        setValue(instant);
    }

    public Date getValue() {
        return this.instant;
    }

    public void setValue(Date value) {
        this.instant = value;
    }

    public void serialize(Element parent) {
        Document doc = parent.getOwnerDocument();
        Element e = doc.createElement("AuthenticationInstant");
        e.appendChild(doc.createTextNode(instant.toString()));
        parent.appendChild(e);
    }

    public void deserialize(Element source) {
        String value = SAMLUtil.getInnerText(source);
        instant = new Date(value);
    }
}
```

Example C-28. AuthenticationLocale.java

```
package saml;

import org.w3c.dom.Element;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;

public class AuthenticationLocale implements AuthenticationLocaleType {

    private String ip;
    private String domain;


    public String getIP() {
        return this.ip;
    }

    public void setIP(String ip) {
        this.ip = ip;
    }

    public String getDNSDomain() {
        return this.domain;
    }

    public void setDNSDomain(String domain) {
        this.domain = domain;
    }

    public void serialize(Element parent) {
        Document doc = parent.getOwnerDocument();
```

```
        Element         e         =         doc.createElementNS(SAMLUtil.NS,
"AuthenticationLocale");
        if (ip != null) {
            Element e1 = doc.createElement("IP");
            e1.appendChild(doc.createTextNode(ip));
            e.appendChild(e1);
        }
        if (domain != null) {
            Element e2 = doc.createElement("DNS_Domain");
            e2.appendChild(doc.createTextNode(domain));
            e.appendChild(e2);
        }
        parent.appendChild(e);
    }

    public void deserialize(Element source) {
        NodeList nl = source.getChildNodes();
        for (int n = 0; n < nl.getLength(); n++) {
            Node node = nl.item(n);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element e = (Element)node;
                if ("IP".equals(e.getLocalName())) {
                    String ip = SAMLUtil.getInnerText(e);
                    setIP(ip);
                }
                if ("DNS_Domain".equals(e.getLocalName())) {
                    String dns = SAMLUtil.getInnerText(e);
                    setDNSDomain(dns);
                }
            }
        }
    }
}
```

Example C-29. AuthenticationLocaleType.java

```
package saml;

public interface AuthenticationLocaleType {

    public String getIP();
    public void setIP(String ip);
    public String getDNSDomain();
    public void setDNSDomain(String domain);

}
```

Example C-30. AuthenticationMethod.java

```
package saml;

import org.w3c.dom.Element;
import org.w3c.dom.Document;

public class AuthenticationMethod {

    private String value;

    public AuthenticationMethod() {}
```

```
    public AuthenticationMethod(String value) {
        setText(value);
    }

    public String getText() {
        return this.value;
    }

    public void setText(String value) {
        this.value = value;
    }

    public void serialize(Element parent) {
        Document doc = parent.getOwnerDocument();
        Element         e         =         doc.createElementNS(SAMLUtil.NS,
"AuthenticationMethod");
        e.appendChild(doc.createTextNode(value));
        parent.appendChild(e);
    }

    public void deserialize(Element source) {
        String s = SAMLUtil.getInnerText(source);
        setText(s);
    }
}
```

Example C-31. IDType.java

```
package saml;

public class IDType {

    private String value;

    public IDType() {}

    public IDType(String value) {
        setText(value);
    }

    public String getText() {
        return this.value;
    }

    public void setText(String value) {
        this.value = value;
    }

}
```

Example C-32. NameIdentifier.java

```
package saml;

import org.w3c.dom.Element;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;

public class NameIdentifier implements NameIdentifierType {
```

```
    private String domain;
    private String name;

    public String getSecurityDomain() {
        return this.domain;
    }

    public void setSecurityDomain(String securityDomain) {
        this.domain = securityDomain;
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void serialize(Element parent) {
        Document doc = parent.getOwnerDocument();
        Element e = doc.createElementNS(SAMLUtil.NS, "NameIdentifier");
        Element e1 = doc.createElement("SecurityDomain");
        e1.appendChild(doc.createTextNode(domain));
        e.appendChild(e1);
        Element e2 = doc.createElement("Name");
        e2.appendChild(doc.createTextNode(name));
        e.appendChild(e2);
        parent.appendChild(e);
    }

    public void deserialize(Element source) {
        NodeList nl = source.getChildNodes();
        for (int n = 0; n < nl.getLength(); n++) {
            Node node = nl.item(n);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element e = (Element)node;
                if ("SecurityDomain".equals(e.getLocalName())) {
                    String sd = SAMLUtil.getInnerText(e);
                    setSecurityDomain(sd);
                }
                if ("Name".equals(e.getLocalName())) {
                    String name = SAMLUtil.getInnerText(e);
                    setName(name);
                }
            }
        }
    }
}
```

Example C-33. NameIdentifierType.java

```
package saml;

public interface NameIdentifierType {

    public String getSecurityDomain();
    public void setSecurityDomain(String securityDomain);
    public String getName();
    public void setName(String name);
```

```
}
```

Example C-34. SAMLUtil.java

```
package saml;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;
import org.w3c.dom.Document;
import org.w3c.dom.Node;

public class SAMLUtil {

    public static final String NS =
      "http://www.oasis-open.org/committees/security/docs/draft-sstc-
schema-assertion-15.xsd";

    public static String getInnerText(Node e) {
        NodeList nl = e.getChildNodes();
        StringBuffer strbuf = new StringBuffer();
        for (int n = 0; n < nl.getLength(); n++) {
            Node node = nl.item(n);
            if (node.getNodeType() == Node.TEXT_NODE) {
                strbuf.append(node.getNodeValue());
            } else {
                strbuf.append(getInnerText(node));
            }
        }
        return strbuf.toString();
    }

    public static Document newDocument() {
        try {
            DocumentBuilderFactory                    dbf                    =
DocumentBuilderFactory.newInstance();
            dbf.setValidating(false);
            dbf.setNamespaceAware(true);
            DocumentBuilder db = dbf.newDocumentBuilder();
            return db.newDocument();
        } catch (Exception e) {
            return null;
        }
    }

}
```

Example C-35. Subject.java

```
package saml;

import java.util.List;
import java.util.Vector;
import java.util.Iterator;
import org.w3c.dom.Element;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
```

```
public class Subject implements SubjectType {

    private List nameid = new Vector();

    public NameIdentifier getNameIdentifier(int index) {
        return (NameIdentifier)this.nameid.get(index);
    }

    public void setNameIdentifier(NameIdentifier nameIdentifier) {
        this.nameid.add(nameIdentifier);
    }

    public void serialize(Element parent) {
        Document doc = parent.getOwnerDocument();
        Element e = doc.createElementNS(SAMLUtil.NS, "Subject");
        for (Iterator i = nameid.iterator(); i.hasNext();) {
            NameIdentifier ni = (NameIdentifier)i.next();
            ni.serialize(e);
        }
        parent.appendChild(e);
    }

    public void deserialize(Element source) {
        NodeList nl = source.getElementsByTagName("NameIdentifier");
        for (int n = 0; n < nl.getLength(); n++) {
            Element e = (Element)nl.item(n);
            NameIdentifier ni = new NameIdentifier();
            ni.deserialize(e);
            setNameIdentifier(ni);
        }
    }
}
```

Example C-36. SubjectAssertion.java

```
package saml;

import org.w3c.dom.Element;

public abstract class SubjectAssertion
    extends Assertion implements SubjectAssertionAbstractType {

    private Subject subject;

    public Subject getSubject() {
        return this.subject;
    }

    public void setSubject(Subject subject) {
        this.subject = subject;
    }

    protected void serializeSubject(Element e) {
        subject.serialize(e);
    }
}
```

Example C-37. SubjectAssertionAbstractType.java

```
package saml;

public interface SubjectAssertionAbstractType extends AssertionAbstractType
{

    public Subject getSubject();
    public void setSubject(Subject subject);

}
```

Example C-38. SubjectType.java

```
package saml;

public interface SubjectType {

    public NameIdentifier getNameIdentifier(int index);
    public void setNameIdentifier(NameIdentifier nameIdentifier);

}
```

C.8 Codeshare

Example C-39. CodeShareOwner.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="CodeShare_Interfaces"
                            targetNamespace="urn:CodeShare_Interfaces"
                            xmlns:tns="urn:CodeShare_Interfaces"

xmlns:types="urn:CodeShare_Interfaces:DataTypes"
                            xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

    <wsdl:types>
     <xsd:schema version="1.0"
               targetNamespace="urn:CodeShare_Interfaces:DataTypes"
               elementFormDefault="qualified"
               attributeFormDefault="unqualified"
               xmlns:se="http://schemas.xmlsoap.org/soap/encoding/"
               xmlns:xsd="http://www.w3.org/2000/10/XMLSchema" >
       <xsd:import  namespace="http://schemas.xmlsoap.org/soap/encoding/"
             schemaLocation="http://schemas.xmlsoap.org/soap/encoding/"/>
       <xsd:element name="item">
        <xsd:complexType>
         <xsd:sequence>
          <xsd:all>
           <xsd:element name="path" type="xsd:string"
                  nullable="true" minOccurs="0"/>
           <xsd:element name="title" type="xsd:string"
                  nullable="true" minOccurs="0"/>
           <xsd:element name="fullpath" type="xsd:string"
                  nullable="true" minOccurs="0"/>
           <xsd:element name="type" type="xsd:string"
                  nullable="true" minOccurs="0"/>
          </xsd:all>
          <xsd:any namespace='xmlns:dc="http://purl.org/dc/elements/1.1/"'
                  processContents="lax" minOccurs="0"
                  maxOccurs="unbounded"/>
         </xsd:sequence>
        </xsd:complexType>
```

```
   </xsd:element>
   <xsd:complexType name="ArrayOfItems">
    <xsd:annotation>
     <xsd:documentation>
       Array of CodeShare item elements
     </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
     <xsd:extension base="se:Array">
      <xsd:attribute ref="se:arrayType"
                     wsdl:arrayType="types:item[]" />
     </xsd:extension>
    </xsd:complexContent>
   </xsd:complexType>
  </xsd:schema>
</wsdl:types>

<wsdl:message name="search">
    <part name="p1" type="xsd:string" />
    <part name="p2" type="xsd:string" />
</wsdl:message>
<wsdl:message name="searchResponse">
    <part name="response" type="types:ArrayOfItems" />
</wsdl:message>

<wsdl:message name="get">
    <part name="p1" type="xsd:string" />
    <part name="p2" type="xsd:string" />
</wsdl:message>
<wsdl:message name="getResponse">
    <part name="response" type="types:ArrayOfItems" />
</wsdl:message>

<wsdl:message name="info">
    <part name="p1" type="xsd:string" />
    <part name="p2" type="xsd:string" />
</wsdl:message>
<wsdl:message name="infoResponse">
    <part name="response" type="types:ArrayOfItems" />
</wsdl:message>

<wsdl:message name="list">
    <part name="p1" type="xsd:string" />
    <part name="p2" type="xsd:string" />
</wsdl:message>
<wsdl:message name="listResponse">
    <part name="response" type="types:ArrayOfItems" />
</wsdl:message>

<wsdl:portType name="CodeShareOwnerInterface">
    <wsdl:operation name="search" parameterOrder="p1 p2">
        <wsdl:input name="search" message="tns:search" />
        <wsdl:output name="searchResponse"
                     message="tns:searchResponse" />
    </wsdl:operation>
    <wsdl:operation name="get" parameterOrder="p1 p2">
        <wsdl:input name="search" message="tns:search" />
        <wsdl:output name="searchResponse"
                     message="tns:searchResponse" />
    </wsdl:operation>
    <wsdl:operation name="info" parameterOrder="p1 p2">
```

```
            <wsdl:input name="search" message="tns:search" />
            <wsdl:output name="searchResponse"
                         message="tns:searchResponse" />
        </wsdl:operation>
        <wsdl:operation name="list" parameterOrder="p1 p2">
            <wsdl:input name="search" message="tns:search" />
            <wsdl:output name="searchResponse"
                         message="tns:searchResponse" />
        </wsdl:operation>
    </wsdl:portType>

    <wsdl:binding name="CodeShareOwner_SOAP_HTTP"
                  type="tns:CodeShareOwnerInterface">

      <soap:binding style="rpc"
                    transport="http://schemas.xmlsoap.org/soap/http" />

      <wsdl:operation name="search">
       <soap:operation soapAction="urn:CodeShareOwner#search" />
       <wsdl:input>
        <soap:body use="encoded" namespace="urn:CodeShareOwner"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
       </wsdl:input>
       <wsdl:output name="Name">
         <soap:body use="encoded" namespace="urn:CodeShareOwner"
              encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
       </wsdl:output>
     </wsdl:operation>

     <wsdl:operation name="get">
      <soap:operation soapAction="urn:CodeShareOwner#get" />
      <wsdl:input>
        <soap:body use="encoded" namespace="urn:CodeShareOwner"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </wsdl:input>
      <wsdl:output>
       <soap:body use="encoded" namespace="urn:CodeShareOwner"
                  encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
/>
      </wsdl:output>
     </wsdl:operation>
     <wsdl:operation name="info">
      <soap:operation soapAction="urn:CodeShareOwner#info" />
      <wsdl:input>
        <soap:body use="encoded" namespace="urn:CodeShareOwner"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </wsdl:input>
      <wsdl:output>
       <soap:body use="encoded" namespace="urn:CodeShareOwner"
                  encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="list">
     <soap:operation soapAction="urn:CodeShareOwner#list"/>
     <wsdl:input>
      <soap:body use="encoded" namespace="urn:CodeShareOwner"
```

```
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
/>
     </wsdl:input>
     <wsdl:output>
      <soap:body use="encoded" namespace="urn:CodeShareOwner"
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
/>
     </wsdl:output>
   </wsdl:operation>
 </wsdl:binding>
</wsdl:definitions>


</wsdl:definitions>
```

Example C-40. AuthenticationService.java

```
package codeshare;

import org.w3c.dom.Element;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import saml.*;

public class AuthenticationService {

    private static String users = "users.xml";
    private static Document doc;
    static {
        doc = XMLUtil.get(users);
        if (doc == null) {
            doc = SAMLUtil.newDocument();
            Element u = doc.createElement("users");
            doc.appendChild(u);
            XMLUtil.put(users, doc);
        }
    }

    public static boolean register(String userid, String password) {
        Element e = doc.getDocumentElement();
        NodeList nl = e.getElementsByTagName("user");
        for (int n = 0; n < nl.getLength(); n++) {
            Element ex = (Element)nl.item(n);
            if (ex.getAttribute("id").equals(userid)) {
                throw new IllegalArgumentException("A user with that ID
already exists!");
            }
        }
        Element u = doc.createElement("user");
        u.setAttribute("id", userid);
        u.setAttribute("password", password);
        e.appendChild(u);
        XMLUtil.put(users, doc);
        return true;
    }

    public static Element login(String userid, String password)
        throws Exception {
        Element el = doc.getDocumentElement();
        NodeList nl = el.getElementsByTagName("user");
        for (int n = 0; n < nl.getLength(); n++) {
```

```
            Element e = (Element)nl.item(n);
            if (e.getAttribute("id").equals(userid) &&
                e.getAttribute("password").equals(password)) {

                AuthenticationAssertion aa = AssertionFactory.newInstance(
                    new String(new Long(
                        System.currentTimeMillis()).toString()),
                        "CodeShare.org",
                        new java.util.Date(),
                        userid,
                        "CodeShare.org",
                        "http://codeshare.org",
                        new java.util.Date(),
                        java.net.InetAddress.
                            getLocalHost().getHostAddress(),
                            java.net.InetAddress.
                                getLocalHost().getHostName());

                Element sa = AssertionSigner.sign(aa, "CodeShare.db",
                             "CodeShare", "CodeShare", "CodeShare");
                return sa;
            }
        }
        return null;
    }

}
```

Example C-41. Authentication Service Deployment Descriptor

```
<isd:service xmlns:isd="http://xml.apache.org/xml-soap/deployment"
             id="urn:CodeShareService-ClientService">
  <isd:provider type="java"
                scope="Application"
                methods="register login">
    <isd:java class="codeshare.AuthenticationService"/>
  </isd:provider>
  <isd:faultListener>org.apache.soap.server.DOMFaultListener
  </isd:faultListener>
</isd:service>
```

Example C-42. VerificationService.java

```
package codeshare;

import org.w3c.dom.Element;
import com.ibm.xml.dsig.*;
import java.security.Key;

public class VerificationService {

    public static boolean isValid(Element signature) throws Exception {

        Key key = null;
        Element keyInfoElement = KeyInfo.searchForKeyInfo(signature);
        if (keyInfoElement != null) {
            KeyInfo keyInfo = new KeyInfo(keyInfoElement);
            key = keyInfo.getKeyValue();
        }
        SignatureContext context = new SignatureContext();
```

```
        Validity validity = context.verify(signature, key);
        return validity.getCoreValidity();
    }

}
```

Example C-43. Verification Service Deployment Descriptor

```
<isd:service xmlns:isd="http://xml.apache.org/xml-soap/deployment"
             id="urn:CodeShareService-Verification">
  <isd:provider type="java"
                scope="Application"
                methods="verify">
    <isd:java class="codeshare.VerificationService"/>
  </isd:provider>
  <isd:faultListener>org.apache.soap.server.DOMFaultListener
  </isd:faultListener>
</isd:service>
```

Example C-44. MasterIndexService.java

```
package codeshare;

import org.w3c.dom.Element;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import saml.*;

/**
 * Master Index Service
 */

public class MasterIndexService {

    private static String owners = "owners.xml";
    private static Document doc;
    static {
        doc = XMLUtil.get(owners);
        if (doc == null) {
            doc = SAMLUtil.newDocument();
            Element u = doc.createElement("owners");
            doc.appendChild(u);
            XMLUtil.put(owners, doc);
        }
    }

    public static boolean register(String ownerid, String password, String
url) {
        Element e = doc.getDocumentElement();
        NodeList nl = e.getElementsByTagName("owner");
        for (int n = 0; n < nl.getLength(); n++) {
            Element ex = (Element)nl.item(n);
            if (ex.getAttribute("id").equals(ownerid)) {
                throw new IllegalArgumentException("An owner with that ID
already exists!");
            }
        }
        Element u = doc.createElement("owner");
        u.setAttribute("id", ownerid);
```

```
        u.setAttribute("password", password);
        u.setAttribute("url", url);
        e.appendChild(u);
        XMLUtil.put(owners, doc);
        return true;
    }

    public static boolean login(String ownerid, String password, Element
index) {
        Element el = doc.getDocumentElement();
        NodeList nl = el.getElementsByTagName("owner");
        for (int n = 0; n < nl.getLength(); n++) {
            Element e = (Element)nl.item(n);
            if (e.getAttribute("id").equals(ownerid) &&
                e.getAttribute("password").equals(password)) {
                Element i = (Element)doc.importNode(index, true);
                NodeList c = e.getElementsByTagName("index");
                if (c.getLength() > 0) {
                    Node node = c.item(1);
                    e.replaceChild(node, i);
                } else {
                    e.appendChild(i);
                }
                XMLUtil.put(owners, doc);
                return true;
            }
        }
        return false;
    }

    public static boolean update(String ownerid, String password,
                                 Element index) {
        Element el = doc.getDocumentElement();
        NodeList nl = el.getElementsByTagName("owner");
        for (int n = 0; n < nl.getLength(); n++) {
            Element e = (Element)nl.item(n);
            if (e.getAttribute("id").equals(ownerid) &&
                e.getAttribute("password").equals(password)) {
                Element i = (Element)doc.importNode(index, true);
                NodeList c = e.getElementsByTagName("index");
                if (c.getLength() > 0) {
                    Node node = c.item(1);
                    e.replaceChild(node, i);
                } else {
                    e.appendChild(i);
                }
                XMLUtil.put(owners, doc);
                return true;
            }
        }
        return false;
    }

}
```

Example C-45. Master Index Service Deployment Descriptor

```
<isd:service xmlns:isd="http://xml.apache.org/xml-soap/deployment"
             id="urn:CodeShareService-MasterIndex">
  <isd:provider type="java"
                scope="Application"
                methods="register update">
    <isd:java class="codeshare.IndexService"/>
  </isd:provider>
  <isd:faultListener>org.apache.soap.server.DOMFaultListener
  </isd:faultListener>
</isd:service>
```

Example C-46. OwnerService.java

```
package codeshare;

import org.apache.regexp.RE;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;
import saml.SAMLUtil;

public class OwnerService {
  private static String index = "index.xml";
  private static org.w3c.dom.Document doc;
  static {
    doc = XMLUtil.get(index);
    if (doc == null)
    {
      doc = SAMLUtil.newDocument();
      Element e = doc.createElement("index");
      doc.appendChild(e);
      XMLUtil.put(index, doc);
    }
  }

  public org.w3c.dom.Element search(String p1) {
    return search(p1, "dc:Title");
  }

  public Element search(String p1, String p2)
  {
    Element e = doc.getDocumentElement();
    NodeList nl = e.getElementsByTagName(p2);

    Document d = SAMLUtil.newDocument();
    Element list = doc.createElement("list");
    d.appendChild(list);

    for (int n = 0; n < nl.getLength(); n++)
    {
      Element next = (Element)nl.item(n);
      try
      {
        RE targetRE = new RE(p1);
        if (targetRE.match(SAMLUtil.getInnerText(next.getText())))
        {
          Element item = (Element)d.importNode(next);
          list.appendChild(item);
        }
      }
      catch (Exception exc) {}
```

```
    }
    return list;
  }


  public Element list(String p1)
  {
    return search(p1, "dc:Title");
  }

  public Element list(String p1, String p2)
  {
    Element e = doc.getDocumentElement();
    NodeList nl = e.getElementsByTagName(p2);

    Document d = SAMLUtil.newDocument();
    Element list = doc.createElement("list");
    d.appendChild(list);

    for (int n = 0; n < nl.getLength(); n++)
    {
      Element next = (Element)nl.item(n);
      try
      {
        RE targetRE = new RE(p1);
        if (targetRE.match(SAMLUtil.getInnerText(next.getText())))
        {
          Element item = (Element)d.importNode(next);
          list.appendChild(item);
        }
      }
      catch (Exception exc) {}
    }
    return list;
  }

  public Element info(String p1) {
    throw new IllegalArgumentException("Not Implemented");
  }

  public Element get(String p1) {
    throw new IllegalArgumentException("Not Implemented");
  }
}
```

Example C-47. Owner Service Deployment Descriptor

```
<isd:service xmlns:isd="http://xml.apache.org/xml-soap/deployment"
             id="urn:CodeShareService-OwnerService">
  <isd:provider type="java"
                scope="Application"
                methods="list search">
    <isd:java class="codeshare.OwnerService"/>
  </isd:provider>
  <isd:faultListener>org.apache.soap.server.DOMFaultListener
  </isd:faultListener>
</isd:service>
```

Example C-48. XMLUtil.java

```
package codeshare;

import java.io.FileWriter;
import javax.xml.parsers.*;
import org.w3c.dom.*;
import org.apache.xml.serialize.*;

public class XMLUtil {

    public static Document get(String path) {
        try {
            DocumentBuilderFactory                    dbf                    =
DocumentBuilderFactory.newInstance();
            dbf.setValidating(false);
            dbf.setNamespaceAware(true);
            DocumentBuilder db = dbf.newDocumentBuilder();
            return db.parse(path);
        } catch (Exception e) {
            return null;
        }
    }

    public synchronized static boolean put(String path, Document doc) {
        try {
            FileWriter fw = new FileWriter(path);
            OutputFormat of = new OutputFormat();
            of.setIndenting(true);
            XMLSerializer x = new XMLSerializer(fw, of);
            x.serialize(doc);
            fw.close();
            return true;
        } catch (Exception e) {
            return false;
        }
    }

}
```

Example C-49. Codeshare/Owner.pm

```
package CodeShare::Owner;

use strict;

my $index;                                          # parsed index file
my $DC_NS = "http://purl.org/dc/elements/1.1/"; # Dublin Code namespace
my @ELEMENTS = qw(Title Creator Date Subject Description);

sub init {
  my($class, $root) = @_;
  open(F, $root) or die "$root: $!\n";
  $index   =   SOAP::Custom::XML::Deserializer->deserialize(join   '',   <F>)-
>root;
  close(F) or die "$root: $!\n";
}

sub traverse {
  my($self, %params) = @_;
```

```
  my $start = $params{start};

  my $type = $start->SOAP::Data::name; # file|project|directory
  my $location = ref $start->location ? $start->location->value : '';

  # path to current structure. Empty for projects
  my $path = $type eq 'directory' ||
             $type eq 'file' ? join('/', $params{path} || (), $location) :
'';
  my $prefix = $type eq 'project' ? $location : $params{prefix} || '';
  my $fullpath = join '/', $prefix, $path; # full path. Used to GET files

  my $where = $params{where};
  my $matched =
    $params{get} && $params{matched} ||
    $params{what} &&
    # check only subelements in Dublin Core namespace
    $start->$where() =~ /$params{what}/ && $start->$where()->uri eq $DC_NS;

  return
    # current element
    ($matched
       ? +{ type => $type,
            path => $path,
            ($params{get} ? (fullpath => $fullpath) : ()),
            map { ref $start->$_() ? ($_ => $start->$_()->value) : ()
                } @ELEMENTS
          }
       : ()
    ),

    # and everything below
    map { $self->traverse(start => $_, where => $where, what =>
$params{what},
                          path => $path, prefix => $prefix,
                          get => ($params{get} || 0), matched => $matched)
}
      $start->project, $start->directory, ($type eq 'file' ? () : $start-
>file)
  ;
}

sub list {

  print("\nHandling a list request...");

  my($self, $what) = @_;

  [ map { my $e = $_; +{ map {$_ => $e->{$_}} qw(type path Title file
fullpath) } }
      $self->traverse(start => $index, where => 'Title', what => $what, get
=> 1)
  ];
}

sub get {

  print("\nHandling a get request...");

  my $results = shift->list(@_);
```

```
  [ map { $_->{type} eq 'file' && open(F, delete $_->{fullpath})
            ? ($_->{file} = join('', <F>), close F) : (); $_ }
      @$results
  ];
}

sub search { # same as info(), but returns only 'type', 'path' and 'Title'

  print("\nHandling a search request...");

  my $results = shift->info(@_);

  [ map { my $e = $_; +{ map {$_ => $e->{$_}} qw(type path Title) } }
      @$results
  ];
}

sub info {

  print("\nHandling an info request...");

  my($self, $what, $where) = @_;

  [ $self->traverse(start => $index,
                    where => $where || 'Title', what => $what || '.')
  ];
}

1;
```

Example C-50. Codeshare.pl (standalone HTTP Daemon)

```
#!perl -w
#!d:\perl\bin\perl.exe

use SOAP::Transport::HTTP;
use CodeShare::Owner;


print "\n\nWelcome to CodeShare! The Open source code sharing network!";
print   "\nCopyright(c)   2001,   James   Snell,   Pavel   Kulchenko,   Doug
Tidwell\n\n";


CodeShare::Owner->init(shift or die "Usage: $0 <path/to/index.xml>\n");

my $daemon = SOAP::Transport::HTTP::Daemon
  -> new (LocalPort => 8080)
  -> dispatch_to('CodeShare::Owner::(?:get|search|info|list)')
;
print "CodeShare Owner Server started at ", $daemon->url, "\n";
print "Waiting for a request...\n";
$daemon->handle;
```

Example C-51. Codeshare.cgi (alternative to standalone HTTP daemon)

```
#!/usr/bin/env perl
# -- Copyright (C) 2001 Pavel Kulchenko --

use strict;
```

```
use SOAP::Transport::HTTP;
use CodeShare::Owner;

CodeShare::Owner->init('../Projects/index.xml');

my $daemon = SOAP::Transport::HTTP::CGI
  -> dispatch_to('CodeShare::Owner::(?:get|search|info|list)')
  -> handle;
;
```

Example C-52. Startserver.bat

```
@echo off

start "CodeShare Owner Server" perl cs_server.pl ..\Projects\index.xml
```

Example C-53. Startserver.sh

```
perl cs_server.pl ../Projects/index.xml
```

Example C-54. Codeshare_client.pl

```
#!/bin/env perl
#!d:\perl\bin\perl.exe

use strict;
use SOAP::Lite;
use File::Path;

print "\n\nWelcome to CodeShare! The Open source code sharing network!";
print   "\nCopyright(c)   2001,   James   Snell,   Pavel   Kulchenko,   Doug
Tidwell\n\n";

@ARGV or die "Usage: $0 CodeShareServer [commands...] [-dump [filename]]
\n";
my $proxy = shift;
my $uri = 'http://namespaces.soaplite.com/CodeShare/Owner';
my $soap = SOAP::Lite->proxy($proxy)->uri($uri)->on_fault(sub{});

my($dump, $file) = @ARGV > 0 && @ARGV[-1] eq '-dump' ? splice(@ARGV, -1, 1)
:
                   @ARGV > 1 && @ARGV[-2] eq '-dump' ? splice(@ARGV, -2, 2)
:
                   (undef, undef);
if ($dump) {
  print STDERR "Wiredumps are logged in '$file'\n" if $file;
  $file ||= '&STDOUT';                        # STDOUT by default
  open(F, ">>$file") or die "$file: $!\n"; # open in append mode
  select((select(F), $|=1)[0]);               # select non-buffered output
  $soap->on_debug(sub{print F @_});           # debug goes there
  eval "END { close F }";                     # close handle when we are done
}

print STDERR "Usage: { search | info | get | list | quit | help }
[parameters...]\n> ";

while (defined($_ = shift || <>)) {
  next unless /\w/;
  my($method,             $modifier,             $parameters)             =
m!^\s*(\w+)(?:\s*/(\w*)\s)?\s*(.*)!;
```

```
  last if $method =~ /^q(?:uit)?$/i;
  help(), next if $method =~ /^h(?:elp)?$/i;

  my $res = eval "\$soap->$method('$parameters', '$modifier')";

  # check for errors
  $@                            and print(STDERR join "\n", $@, ''), next;
  defined($res)  &&  $res->fault    and  print(STDERR  join  "\n",  $res-
>faultstring, ''), next;
  !$soap->transport->is_success   and   print(STDERR   join   "\n",   $soap-
>transport->status, ''), next;

  # check for result
  my @result = @{$res->result} or print(STDERR "No matches\n"), next;

  foreach (@result) {
    print(STDERR "$_->{type}: @{[join ', ', $_->{Title} || (), $_->{path}
|| ()]}\n");
    if ($method eq 'get') {
      if ($_->{type} eq 'directory') { File::Path::mkpath($_->{path}) }
      if ($_->{type} eq 'file') {
        open(F, '>'. $_->{path}) or warn "$_->{path}: $!\n";
        print F $_->{file};
        close(F) or warn "$_->{path}: $!\n";
      }
    } elsif ($method eq 'info') {
      foreach my $key (grep {$_ !~ /^(?:type|path)/} keys %$_) {
        print "  $key: $_->{$key}\n";
      }
    }
  }
} continue {
  print STDERR "\n> ";
}

sub help {
  print "Short help about search, info, get and list commands is here\n";
}
```

Colophon

Our look is the result of reader comments, our own experimentation, and feedback from distribution channels. Distinctive covers complement our distinctive approach to technical topics, breathing personality and life into potentially dry subjects.

The animal on the cover of *Programming Web Services with SOAP* is a sea sponge. There are thousands of species of sponge (*Phylum Porifera*). Sponges are simple, multicellular animals that feed and breathe by filtering water. They are covered with tiny pores called ostia, which lead to an internal system of canals coated with sticky cells called choanocytes, or collar cells. These cells facilitate water through the canals with constantly moving flagella, picking up oxygen and pieces of food, and carrying out carbon dioxide and waste. The water passes out of the sponge through larger pores called oscula.

Free-standing and encrusting sea sponges live at the bottom of the ocean, in deep and shallow waters. Free-standing sponges can grow to gigantic sizes, and crab, shrimp, sea slugs, and starfish are often found living inside. Encrusting sponges attach themselves to rocks, shells, wood, and kelp. Some sponges produce toxic chemicals, possibly to give them a bad taste to predators. Other sponges have sharp, prickly spines as their only defense.

Colleen Gorman was the production editor and copyeditor for *Programming Web Services with SOAP*. Linley Dolby and Matt Hutchinson provided quality control. Phil Dangler and Camilla Ammirati provided production support. John Bickelhaupt wrote the index.

Ellie Volckhausen designed the cover of this book, based on a series design by Edie Freedman. The cover image is an original illustration created by Susan Hart. Emma Colby produced the cover layout with QuarkTMXPress 4.1 using Adobe's ITC Garamond font.

Melanie Wang designed the interior layout, based on a series design by David Futato. Neil Walls converted the files from Microsoft Word to FrameMaker 5.5.6 using tools created by Mike Sierra. The text font is Linotype Birka; the heading font is Adobe Myriad Condensed; and the code font is LucasFont's TheSans Mono Condensed. The illustrations that appear in the book were produced by Robert Romano and Jessamyn Read using Macromedia FreeHand 9 and Adobe Photoshop 6. This colophon was written by Colleen Gorman.